AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated in the following listing of all claims:

1. – 2. (Cancelled)

3. (Previously Presented) The method of claim 31, wherein the first and second instructions are executed in parallel.

4. (Previously Presented) The method of claim 31, wherein the first instruction and the second instruction correspond to a very long instruction word packet (VLIW).

5. – 7. (Cancelled)

8. (Previously Presented) The method of claim 31, wherein the second instruction includes an opcode field, fields to indicate at least a first and second operands, and a destination field to indicate where to store the carry condition indication.

9. (Cancelled)

10. (Previously Presented) The method of claim 31 embodied in a computer program product that is encoded on one or more machine-readable media.

11. (Cancelled)

12. (Previously Presented) The method of claim 36, wherein a very long instruction word packet includes at least one of the generate carry instruction instances.

13. (Previously Presented) The method of claim 36, wherein at least some of the generate carry instruction instances are performed in parallel with corresponding instances of addition type instructions.

14. – 16. (Cancelled)

17. (Previously Presented) The processor of claim 39 that also supports separately executable subtraction type and generate borrow instructions, wherein an instance of the generate borrow instruction indicates at least one of the store units to store an indication of a borrow condition that corresponds to an instance of the subtract type instruction.

18. (Previously Presented) The processor of claim 39 that supports parallel execution of the separately executable instructions.

19. (Previously Presented) The processor of claim 39 that decodes the separately executable instructions from a very long instruction word packet.

20. (Cancelled)

21. (Previously Presented) The apparatus of claim 41 further comprising means for indicating a borrow bit in the store unit responsive to execution of a third instruction that determines whether subtraction of operands will cause a borrow condition separate from execution of a fourth instruction that subtracts the operands.

22. (Previously Presented) The apparatus of claim 21 further comprising means for executing instances of the third and fourth instructions in parallel.

23. (Cancelled)

24. (Previously Presented) The apparatus of claim 41 further comprising means for performing instances of the first and second instructions in parallel.

25. – 26. (Cancelled)

27. (Previously Presented) The computer program product of claim 43 wherein the generate carry instruction includes an opcode field, a plurality of source operand fields, and a destination field.

28. (Previously Presented) The computer program product of claim 43, wherein the instances of the generate carry and the addition type instructions are executable on separate processing units.

29. (Previously Presented) The computer program product of claim 43, wherein the instances of the generate carry and the addition type instructions are executable along separate paths in parallel.

30. (Cancelled)

31. (Previously Presented) A method comprising:

executing a first instruction to perform an addition type arithmetic operation on at least a first and second operands; and

executing a second instruction to indicate whether the addition type arithmetic operation causes a carry condition, wherein the second instruction includes a field that indicates a destination location for the carry condition indication.

32. (Previously Presented) The method of claim 31 further comprising:

executing a third instruction to perform a subtraction type arithmetic operation on a third and fourth operands; and

executing a fourth instruction to indicate whether the subtraction type arithmetic operation causes a borrow condition, wherein the fourth instruction includes a field that indicates a destination location for the borrow condition indication.

33. (Previously Presented) The method of claim 32 further comprising employing the fourth instruction to determine whether an underflow occurs.

34. (Previously Presented) The method of claim 31 further comprising:

executing a third instruction to perform an addition type arithmetic operation on a third operand, fourth operand, and the result of the second instruction;

executing a fourth instruction to indicate whether the third instruction causes a carry condition, wherein the second instruction includes a field that indicates a destination location for the carry condition indication; and

using the carry condition indication from the fourth instruction to determine whether an overflow occurs from the third instruction,

wherein the first and third operands are respectively a lower and upper portions of a first multi-word value and the second and fourth operands are respectively a lower and upper portions of a second multi-word value.

35. (Previously Presented) The method of claim 34 further comprising:

if an overflow occurs, executing a fifth instruction to add the result of the fourth instruction to the result of the first instruction, and concatenating the result of the fifth instruction to the result of the third instruction; and

if the overflow does not occur, then concatenating the result of the first and third instructions.

36. (Previously Presented) A method comprising:

executing instances of a generate carry instruction to indicate carry conditions for respective sets of operands if additions of the respective sets of operands cause a carry condition,

wherein a first of each of the sets of operands is a word of a multi-word operand.

37. (Currently Amended) The method of claim 36 further comprising using ~~the~~ an instance of a second instruction to determine if an overflow occurs for at least addition of a first of the sets of operands.

38. (Previously Presented) The method of claim 36 further comprising:

executing instances of a generate borrow instruction to indicate borrow conditions for respective sets of operands if subtractions of the respective sets of operands cause a borrow condition.

39. (Previously Presented) A processor that supports a generate carry instruction separate from an addition type instruction and that includes a plurality of store units, wherein execution of an instance of the generate carry instruction causes the processor to store an indication of a carry condition generated by the generate carry instruction in one of the plurality of store units as identified by the generate carry instruction.

40. (Previously Presented) The processor of claim 39, wherein the store units include general registers.

41. (Previously Presented) An apparatus comprising:

a store unit; and

means for indicating a carry bit in the store unit responsive to execution of a first instruction that determines whether addition of operands will cause a carry condition separate from execution of a second instruction that adds the operands.

42. (Previously Presented) The apparatus of claim 41 further comprising means for determining overflow with the second instruction.

43. (Previously Presented) A computer program product encoded on one or more machine-readable media, the computer program product comprising:

a generate carry instruction executable to generate a carry bit if addition of a set of operands causes a carry condition and that includes fields to indicate the set of operands and a destination for the carry bit; and

an addition type instruction executable to add a set of operands.

44. (Previously Presented) The computer program product of claim 43 further comprising:

a generate borrow instruction executable to generate a borrow bit if subtraction of a set of operands causes a borrow condition and that includes fields to indicate the set of operands and a destination for the borrow bit; and

a subtraction type instruction executable to subtract a set of operands.